NASA'S MISSION TO PLANET EARTH

EARTH PROBES

DATA INFORMATION SYSTEM

EOS

EARTH OBSERVING SYSTEM

Modeling Inputs

Mary Armstrong

10 January 1996

Roadmap - Modelling Inputs

- **ECS Technical Baseline (including AHWGP, AHWGC, and user modeling baselines)**

- **Vendor Specifications**

- **Benchmarks**

- **Analysis of custom software to predict its computational and storage requirements**

- **Infrastructure Analysis Assumptions**

Technical Baseline: What?

The ECS Technical Baseline is...

- A centralized collection of load characteristics and descriptions
- Updated approximately four times a year
- Frozen between updates to allow consistent application project-wide

The Technical Baseline is composed of...

- Process and File Descriptions, from AHWGP
- User Access Characteristics, from User Model and AHWGC
- Ancillary Information (e.g., Phasing Factors, V0 Migration info)
- Derived Information (e.g., Processing and Volume Timelines, “X”)
- Design Guidelines (e.g., Use 2X for daily distribution volume)

Technical Baseline: Why, Who?

The Technical Baseline is the source of load information for the ECS design and development staff, and is the primary input to the ECS Dynamic System Model.

The baseline is managed by Joe Guzek (ECS Chief System Architect). Key contributions to its content are provided by the Science Office (AHWGC, AHWGP, SSI&T, User Model).

Uses of the Technical Baseline include:

- Data Server Subsystem development
- Science Data Processing Subsystem development
- Data Management development
- Planning and Scheduling Subsystem development
- Performance (static & dynamic) and other modeling and analyses

This discussion is limited to performance modeling applications.

Technical Baseline: AHWGP

The AHWGP data in the technical baseline provides engineering-relevant descriptions of the anticipated loads associated with Science Software execution.

In particular, the AHWGP data describes files, processes (PGEs), and the logical and quantitative relationships between the two (i.e., what files are input to/output from which processes, and how many of each).

The data form the basis for a variety of analyses:

- Coarse-grained static processing load and archive volume timelines
- “Average” and “Busy day” static load estimates (Mflop/s, I/O, per-process disk usage, etc.)
- Combined analytic queuing network/Petri network model (CAP)
- Discrete event dynamic system simulation

AHWGP: File Descriptions (sample)

File ID	Instrument	File Name	SPSO Equivalent	Archive Site	File Disposition	File Size (MB)	Temporal Coverage (Minutes)	Root / External Flag	Ingest Media Flag	Source
ADEOS-AMSR-L1	AMSR	AMSR L1 Brightness Temp	N/A	JPL	Archive	1750.0	1440.0	2	0	0
ADEOS-ANCILL.	SWS	ADEOS Ancillary Mission Data Products	N/A	JPL	Archive	1.0	1440.0	2	0	0
ADEOS-L0	SWS	Level 0 Raw Data Product	N/A	JPL	Archive	225	1440.0	2	0	0
AGSA_LF_A	Other	Internal Ancillary Data	N/A	LaRC	Interim	22.0	60.0	0	0	5
AGSA_LF_T	Other	Internal Ancillary Data	N/A	LaRC	Interim	22.0	60.0	0	0	5
ANC-NAV	MISR	Platform ancillary data	N0008	LaRC	Archive	0.5	98.88	1	0	5
Anc_AerTau_ThIR	Other/TBD	Data Set needed for Atmospheric Correction of MODIS ThIR Radiances	N/A	GSFC	Interim	5.0	2.46	0	0	3
ANC_AVHRR_NDVI	AVHRR	Global NDVI Product From EDC and GSFC Land Pathfinder Data Sets	none avail.	GSFC	Permanent	300.0	14400.0	0	0	2
ANC_DCW_Land/Sea	Other	Land/Sea Boundary Data	N0030	EDC	Permanent	80.0	500000.0	0	0	2
ANC_EDC_DEM	Other	Digital Elevation Map	N0002	EDC	Permanent	200.0	500000.0	0	0	2
ANC_EDC_LANDCOVER	Other	Surface Land Cover and Vegetation Type	N0006	EDC	Permanent	250.0	129600.0	0	0	2
ANC_ENGDATA_G	MODIS (AM)	MOD02 Input Engineering Data	MOD02	GSFC	Permanent	0.4	500000.0	0	0	3
ANC_EPA_ECOSYSDB	Other	EPA Ecosystems Map 59 Classes at 10 Minute Resolution	N0006	GSFC	Permanent	10.0	500000.0	0	0	0
ANC_ESI	Other	Ancillary Extraterrestrial Solar Irradiance	N0014	GSFC	Permanent	0.5	500000.0	0	0	3
ANC_GEODTM	Other	MOD03 Input Preprocessed Digital Terrain Model	MOD03	GSFC	Permanent	2.7	2.46	0	0	3

AHWGP: Process Descriptions (sample)

Process ID	Process Name	Instrument	Processing Site	Epochs	Input File IDs	# Read per Execution	Amount Read (Fraction)	Output File ID	# Written per Execution	Amt Wrtten (Fraction)	Millions of Floating Point Ops per Execution	No. of Exec. /day	Probability of Execution
AST_PGE_01	Scene classification (stand. processing)	ASTER	EDC	ghijklmnopqrstuvwx	AST_ANC_01	1	1	AST_10	1	1	4,993	182.0	1.0
					AST_ANC_02	1	0.001						
					ANC_EDC_DEM	1	0.001						
					Anc_EDC_LANDCOVER	1	0.001						
					AST_L1B	1	1						
					ANC_DCW_Land/Sea	1	1						
AST_PGE_02	Decorrelation stretch	ASTER	EDC	ghijklmnopqrstuvwx	AST_L1B	1	1	AST_06B	1	1	18,337	182.0	1.0
					AST_10	1	1	AST_06C	1	1			
					AST_DS_TMP1	1	1	AST_06A	1	1			
					AST_DS_TMP2	1	1						
AST_PGE_03	Brightness temperature	ASTER	EDC	ghijklmnopqrstuvwx	AST_ANC_03	1	0.001	AST_04	1	1	861	70.0	0.3846
					AST_L1B	1	1						
					AST_BT_TMP1	1	1						
AST_PGE_04	Atmospheric correction--VNIR, SWIR	ASTER	EDC	ghijklmnopqrstuvwx	MIS-2AS	1	1	AST_09B	1	1	34,200	70.0	0.3846
					MIS-AC	1	1	AST_09A	1	1			
					ANC_EDC_DEM	1	1	AST_07B	1	1			
					AST_ANC_04	1	0.000001	AST_07A	1	1			
					ANC_NMC_PROF	1	0.001						
					AST_ANC_05	1	1						
					ANC_GSFC_O3TOMS	1	1						
					AST_L1B	1	1						
					MOD30_L2_G	1	1						
					AST_04_DEM_GRD_TMP	1	1						
					AST_04_DEM_PIX_TMP	1	1						
					AST_04_MODTN1	1	1						
					AST_04_MODTN2	1	1						

Technical Baseline: non-AHWGP used by model

In addition to the AHWGP, the dynamic model also uses the following information from the Technical Baseline, or derived from it:

- **Definition of “X” for Total Daily Average Distribution Volume**
- **Relative DAAC Access, Service Access Profiles**
- **Diurnal User Request Profile (Requests/minute vs. time of day)**
- **V0 Migration (GB/day vs. DAAC)**
- **Various ICD info (ASF, EDOS, ASTER, TRMM (TSDS/SDPF))**

Technical Baseline: Where to now?

Currently using Aug 95 baseline:

- 102 Process Descriptions
- 426 File Descriptions
- LIS, CERES TRMM/AM/PM, ASTER, MISR, MODIS, MOPITT, DAS (some), SWS, DFA + TSDS, Landsat 7, Version 0, ASF
- MODIS L3s in baseline not modeled

AHWGP update in progress (Due in Jan)

- Number of PDs will probably double
- Number of FDs not likely to change significantly
- Adding SAGE III, DAS (complete), AIRS, GLAS
- Most other Instruments making significant changes to data

Incorporate latest User Model Service Access Profiles in dynamic model

Access patterns from AHWGC

Technical Baseline: What's needed?

- Definition of Reprocessing loads
  - Reprocessing currently modeled as "head of chain".
- Process/File description of SSI&T
  - SSI&T could be modeled as a stochastic mix of single and multiple instances of AHWGP processes with non-causal dependencies.
- User Supplied Methods/Subsetting load characterization
  - Statistical description of load: Ops/byte probability distribution, bytes in/ bytes out probability distribution.
- User Data Access Patterns
  - Space vs. time vs. subset requirements, recent vs. old; all vs. dataset.
- Internals of PGE I/O
- PGE memory requirements
- Browse Product Definition
- SCF Data Flows

Vendor Specifications

Vendor specifications for capacity and performance (H/W) and resource utilization (COTS S/W) are typically a starting point for modeling and sizing analysis.   Examples:

- SGI Power Challenge: current models are rated at 360 MFLOPS
- DCE Client: OSF estimate of resources needed for a DCE client are
    - 24 MB RAM
    - 48 MB disk
- Autosys: RAM requirement = (size of job file * number of jobs * number of GUI displays)

Real experience with COTS hardware and software often provides some adjustment to vendor specifications. This is a primary reason for benchmarking.

Benchmarking

Benchmarking is performed

. . . to provide input data for models

. . . to verify system conformance to requirements

. . . to assist in system sizing

Benchmarking results can be generalized to support modeling efforts (e.g., database benchmarks), can be used to calibrate modeling inputs (e.g., vendor specifications and PGE floating point operations), or can be focused to observe behaviors too complex to cost effectively model.

Benchmarking Infrastructure Components

- Networks: Benchmarks on TCP/IP, FDDI and HiPPI have been performed to determine protocol overheads and effective throughput

- CSS: Benchmarks of DCE-based functions have been performed to determine impact on system performance
  - authentication / authorization
  - OODCE-based inter-process communications
  - client processing requirements

- MSS: Benchmarks of MSS COTS components are being performed to determine
  - overhead requirements for agent and event logging functions
  - MSS COTS requirements for MSS server and workstations (HP Openview, Tivoli, Sybase, Remedy, Clearcase, DDTS, XRP II, Accounting and Billing software)

Benchmarking COTS Products

Hardware and peripherals

- Tape drives - mount, dismount, seek, read, and write
- Robotics - fetch, insert
- RAID - access times and caching performance

COTS Software

- FSMS - internal delays, efficiencies across FSMS products
- File systems (e.g., NFS, AFS) - remote and local access profiles
- Autosys - scheduling performance and overhead, CPU, RAM requirements
- Web server - CPU and RAM requirements

Databases

- Sybase and Illustra
  - response time, CPU and RAM utilization (given straight SQL queries)
  - will use EP6 schema (tailored appropriately for ORDBMS)
  - 1 to 100 simultaneous users, up to 100 GB of metadata

Benchmarking Custom Components

Database storage requirements

- a byproduct of database design
- based on analysis of DID 311

Custom software transaction analysis

- for a given transaction (an element of a thread), analyze the executable instructions
- either estimate or benchmark typical transactions
  - use LOC expansion to machine instructions or
  - use benchmark to calibrate a typical transaction and then generate other estimates by comparison
  - estimate I/O based on interfaces with other software components

Infrastructure Analysis

Modeling of infrastructure will initially focus on services used widely within ECS applications

- Subscription services
- UR generation
- MSS event logging and agent processing
- DCE (client processing, security and name services)
- communications stack (framing overhead, TCP/IP applications)
- e-mail

Benchmarking has been performed or is planned for most infrastructure services.

Parameters for infrastructure functions are based on the technical baseline, user model, and engineering estimates (e.g., number of errors). An Infrastructure Workshop is planned (Feb 5) to discuss this topic in further detail.

Infrastructure: Subscription Assumptions

Parameter: Subscription Notification overhead per day

Function: Overhead$_{\text{subscription}}$ per host per day = Overhead$_{\text{subsciption}}$ *

- **DSS, Ingest -> Planning = ((# of datasets archived + # of QA changes + # of DAS's + # of plans generated) per day) * (# of DAACs + SMC)**
- **DSS, Ingest -> external users = ((# of datasets archived + # QA changes + # of DAS's + # of plans) per day) * # of subscribers**
- **DM -> Planning = (# of metadata adds, deletes, and inserts) per day**
- **DM -> external users = (# of metadata adds, deletes and inserts per day * # of subscribers) + (# of advertising adds, deletes and inserts per day * # of subscribers)**

Rationale:

- **DSS sends subscription notifications regarding time intervals, new data (or version) archived, QA changes, data availability schedules, and plans**
- **DM sends subscription notifications regarding dictionary (metadata) and advertisement updates, inserts and deletes**
- **Ingest sends subsciption notifications regarding new data received.**

Notes: Subscription notification overhead is the computation and I/O required to send or receive a subscription. The current assumption is that subscriptions are used to notify, and not to push data.

Infrastructure:
UR Assumptionss

Parameter: UR creation overhead per host per day

Function: $\text{Overhead}_{\text{UR\_create}}$ per host per day = $\text{Overhead}_{\text{UR\_create}}$ *

DSS: # of data granules requested per day + # of data granules inserted per day + # of subscriptions generated per day + # of sessions established per day

MSS: # of user requests per day

Data Management: # of advertised services added per day

where $\text{Overhead}_{\text{UR\_create}} = \text{I/O\_size}_{\text{UR}}$ or $\text{CPU\_reqt}_{\text{UR\_create}}$

Rationale:

- **URs are used by DSS to point to data, subscriptions, and active/saved sessions.**
- **URs are used by MSS to enable tracking of user requests**
- **URs are used by Data Management to point to advertised services**

Notes: UR creation overhead is the computation and I/O required to create a (unique) UR and translate the application specific location (e.g., path name or database record) to an ascii representation.

Infrastructure:
UR Assumptionss

Parameter: UR_Read overhead per host per day

Function: Overhead$_{\text{UR\_Read}}$ per host per day = Overhead$_{\text{UR\_Read}}$ *

DSS: # of data granules requested from host / day + # of subscriptions accessed from host / day + # of saved sessions re-started or statused from host / day

Data Management: # of advertised services accessed from host / day

MSS: # of user requests / day * # of state changes per request

where Overhead$_{\text{UR\_Read}}$ = I/O_size $_{\text{UR}}$ or CPU_reqt$_{\text{UR\_Read}}$

Rationale:

- **URs are used by users to access data, subscriptions, active/saved sessions, status of user requests, and advertised services**
- **URs are used by DSS and MSS to update status of user requests**
- **URs are used by planning to access data**

Notes: UR read overhead is the computation and I/O required to read a UR and translate the ascii representation to an application specific location (e.g., path name or database record). The computation associated with getting or moving the data at the UR's location is not included.

Infrastructure:
Network Assumptions

Parameter: Communications Stack I/O and processing

Function:

- **TCP/IP over FDDI: 80 Mbps**
- **TCP/IP over HiPPI: 600 Mbps**

Rationale: Benchmarks and published literature

Notes: TCP/IP over HiPPI overhead depends on tuning specific window and buffer sizes. The HiPPI overhead above assumes use of SGI's IRIX 6.2.

Infrastructure: Management Event Logging Assumptions

Parameter: Management event logging overhead (I/O or CPU) per host

Function: Event Logging Overhead per host per day =

Overhead$_{\text{log\_entry}}$ * ((# of errors/day/application * # of applications/host)
+ (# of executed processes or threads / day / host * 2)
+ (# of transaction_events / day / application * # of applications/host)
+ (# of managed applications/host * polling frequency/day)
where Overhead$_{\text{log\_entry}}$ = I/O_size$_{\text{log\_entry}}$ or CPU_reqt$_{\text{log\_entry}}$

Rationale: Logged Events include

- **start and end of processes**
- **managed transactions: archived granules, withdrawn granules, ingested granules, user requests**
- **performance information for major processes: queue lengths, number of active threads, etc.**
- **errors detected**

Notes: The event logger logs both management and application events; the MSS agent only processes management events.

Infrastructure: Management Agent Assumptions

Parameter: Management Agent overhead (I/O or CPU) per host

Function: Management Agent Overhead for one host =

$\text{Overhead}_{\text{agent}}$ * ((# of mgmt_errors/day/application * # of applications/host)
+ (# of executed processes or threads / day / host * 2)
+ (# of mgmt_transaction_events / day / application * # of applications/host)
+ (# of managed applications/host * polling frequency/day)
where $\text{Overhead}_{\text{agent}}$ = $\text{I/O\_size}_{\text{lagent}}$ or $\text{CPU\_reqt}_{\text{agent}}$

Rationale: Logged Events include

- **start and end of processes**
- **managed transactions: archived granules, withdrawn granules, ingested granules, user requests**
- **performance information for major processes: queue lengths, number of active threads, etc.**
- **errors detected**

Notes: The event logger logs both management and application events; the MSS agent only processes management events.

Infrastructure:
DCE Assumptions

Parameter: DCE client processing

Function: The DCE client requirements are

- **24 MB RAM**
- **48 MB disk**
- **CPU -- benchmark analysis is in progress**

Rationale: Published literature and use within EDF

Notes: Directory and security service lookups are assumed to occur in the DCE client, which has sufficient caching to satisfy normal requirement. An exception will be the user login function, in which the ECS client will access the CSS server.

Infrastructure:
E-mail Assumptions

Parameter: E_mail overhead per server per day

Function: Overhead $_{e\_mail}$ per server per day = Overhead $_{e\_mail}$ *
# of external subscription notifications per host per day
where Overhead$_{e\_mail}$ = I/O_size $_{e\_mail\_notify}$ or CPU_reqt $_{send\_mail}$

Rationale:

- **E_mail is assumed to be the normal subscription notification for external requestors**

Notes: This function does not consider e-mail sent by M&O staff from ops workstations.